

- Translate PDU to SDU and Distribution to addressee

NOTE: It is outside the scope of this Standard how the above enumerated steps are implemented as long as an instantiation of the T-KE behaves according to the requirements below.

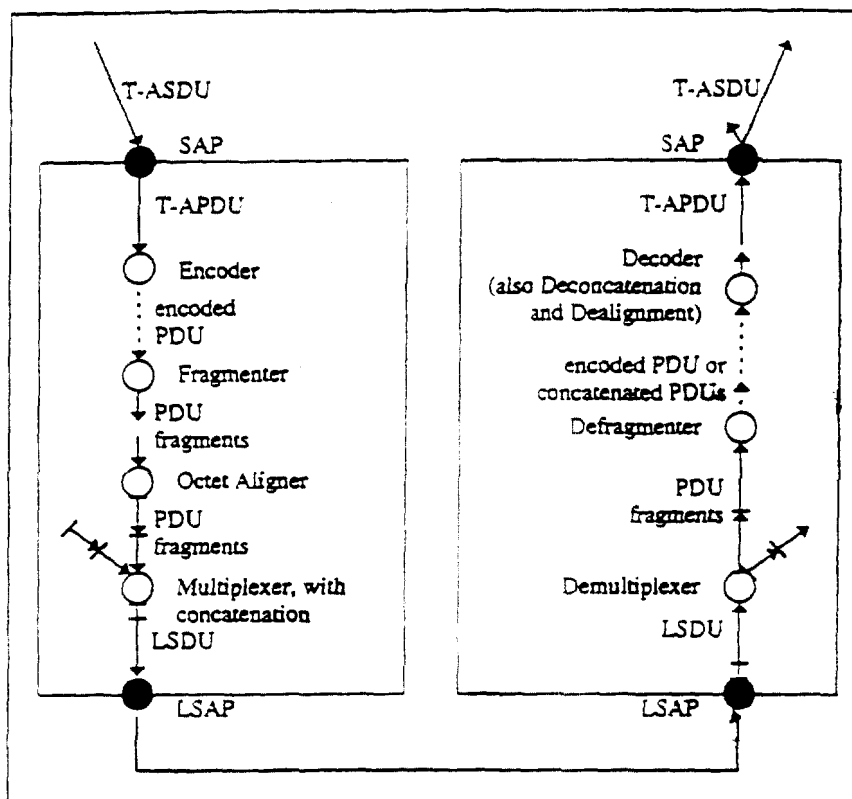


figure 5: Functionalities of the T-KE

6.2.3.1 SDU to PDU

The T-KE shall translate the request and response service primitive into T-APDUs according to the following rules:

A service.request is translated into the corresponding service-request T-APDU defined in Annex A1. A service.response is translated into the corresponding service-response T-APDU defined in Annex A1. The LID shall be removed and shall be given to the LLC in each LLC-service-primitive as defined in 6.2.3.6. In the case of the INITIALISATION.request this LID shall be 11111111₂.

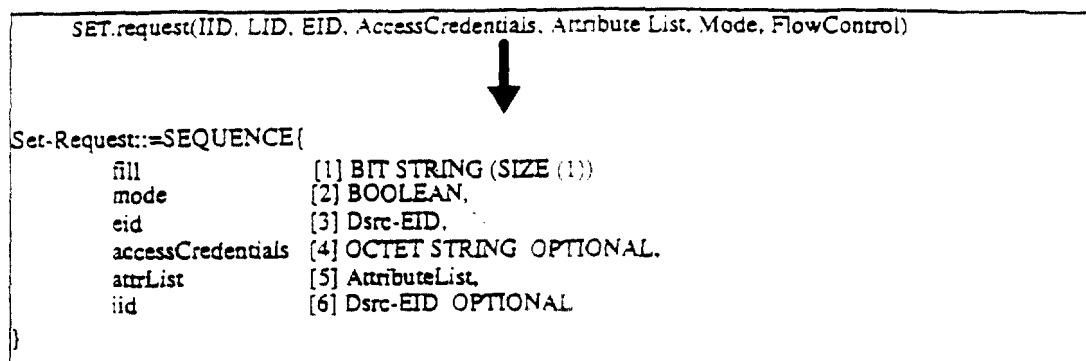


figure 6: SDU to PDU

6.2.3.2 Encoding

The T-KE shall encode the request and response PDUs according to ASN.1-BASIC-PER, UNALIGNED. The encodable ASN.1 types are specified in Annex A.

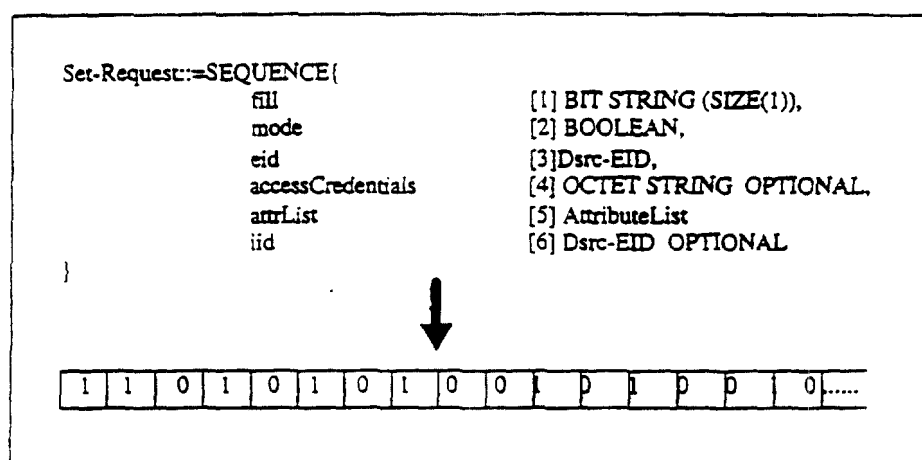


figure 7: Encoding

6.2.3.3 Fragmentation

The T-KE shall fragment encoded PDUs down to T-APDU-fragments, which include a fragmentation header. A fragmentation header shall have a length of at least 1 octet and at most 3 octets. The length of the T-APDU-fragments shall not exceed the maximal LLC packet length. The number of bits inside a fragment shall be a multiple of 8 bits for all but the last fragment. All but the last fragment shall have the same length. Fragmentation shall be made from the most significant bit down to the least significant bit according to ASN.1-BASIC-PER. Fragmentation of multiple PDUs shall be done in parallel.

6.2.3.3.1 Fragmentation header

The first octet of the fragmentation header shall be the first octet of the fragment. If the fragmentation header consists of more than one octet these octets are given in increasing order directly after the first octet.

6.2.3.3.2 Elements of the fragmentation header

The fragmentation header shall consist of one fragmentation indicator, one PDU number, one fragment number, and one fragment number extension indicator. The position of the related bits is described in figure 8. The bits are numbered from 8 to 1 where bit 8 is the most significant and 1 is the least significant bit.

8	7	6	5	4	3	2	1
Fragmentation Indicator	PDU Number				Fragment Counter		Extension indicator

figure 8: One octet fragmentation header

6.2.3.3.3 Fragmentation indicator

The most significant bit (bit 8) of any fragmentation header shall be the fragmentation indicator. The fragmentation indicator shall be 1_2 if this fragment is the last of a sequence of fragments belonging to one PDU or if the PDU is not fragmented. The fragmentation indicator shall be 0_2 if fragmentation is performed and if it is not the last fragmented frame in a message.

6.2.3.3.4 PDU-number

Bit seven to four of the first octet represent a PDU number, which shall be unique for each LID during the defragmentation time in the receiving entities and the same in all T-APDU fragments belonging to one T-APDU. PDU number 0000_2 and 0001_2 shall only be used by T-APDU fragments which are sent by the B-KE.

6.2.3.3.5 One octet fragmentation header

A one octet fragmentation header shall be used if no fragmentation is performed or, if fragmentation is performed, for fragments numbered 0 to 3. A fragment counter shall be used to identify a fragment. Bits 3 and 2 of the first octet shall be interpreted as an unsigned integer, where the highest significant bit is bit 3 of the first octet and the least significant bit is bit 2 of the first octet. Bit 1 shall be set to 1_2 . If fragmentation is performed then the first fragment shall be given the fragment counter value 0, the second fragment the fragment counter value 1 etc.

6.2.3.3.6 Two octet fragmentation header

A two octets fragmentation header shall be used for fragments between 4 and 511, bit 1 of the first octet shall be set to 0_2 . Bits 3 and 2 of the first octet and bit 8 to 2 of the second octet shall be interpreted as an unsigned integer, where the highest significant bit is bit 3 of the first octet and the least significant bit is bit 2 of the second octet. Bit 1 of the second octet shall be set to 1_2 . Fragment numbers shall be assigned to the fragment as specified in 6.2.3.3.5.

6.2.3.3.7 Three octet fragmentation header

A three octets fragmentation header shall be used for fragments between 512 and 65535, bit 1 of the first octet shall be set to 0_2 . Bits 3 and 2 of the first octet, bit 8 to 2 of the second octet and bit 8 to 2 of the third octet are interpreted as unsigned integer, where the highest significant bit is bit 3 of the first octet and the least significant bit is bit 2 of

the third octet. Bit 1 of the second octet shall be set to O_2 and Bit 1 of the third octet shall be set to I_2 . Fragment numbers shall be assigned to the fragment as specified in 6.2.3.3.5.

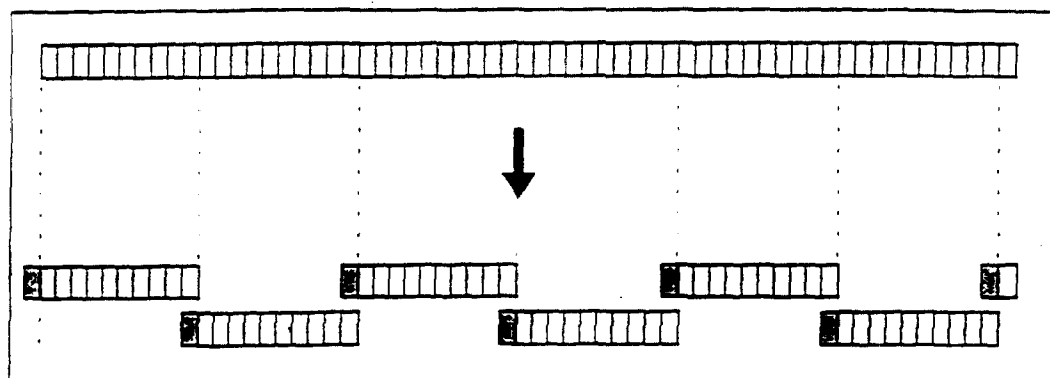


figure 9: Fragmentation

6.2.3.4 Octet Alignment

The T-KE shall add 'zero' bits to a fragment until the number of bits is a multiple of eight. The number of inserted 'zeros' shall be between 0 to 7.

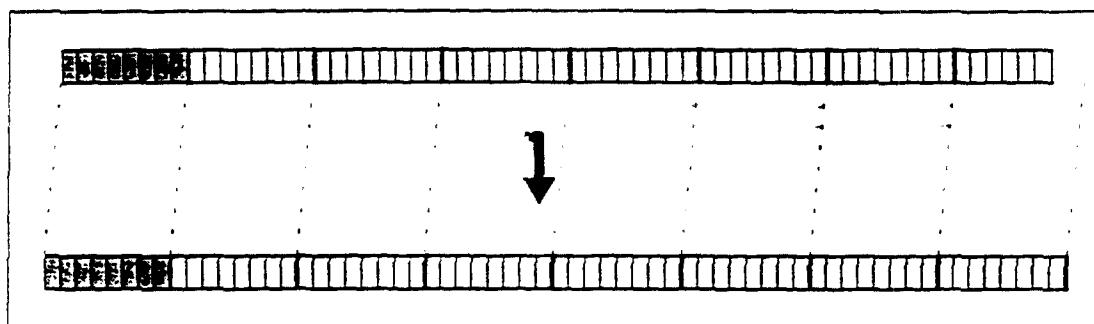


figure 10: Octet Alignment

6.2.3.5 Multiplexing

The T-KE shall multiplex the T-APDU-fragments on the LSAP according to a Head of the line-strategy where the priorities are given by the I-KE (see 6.3.3.2).

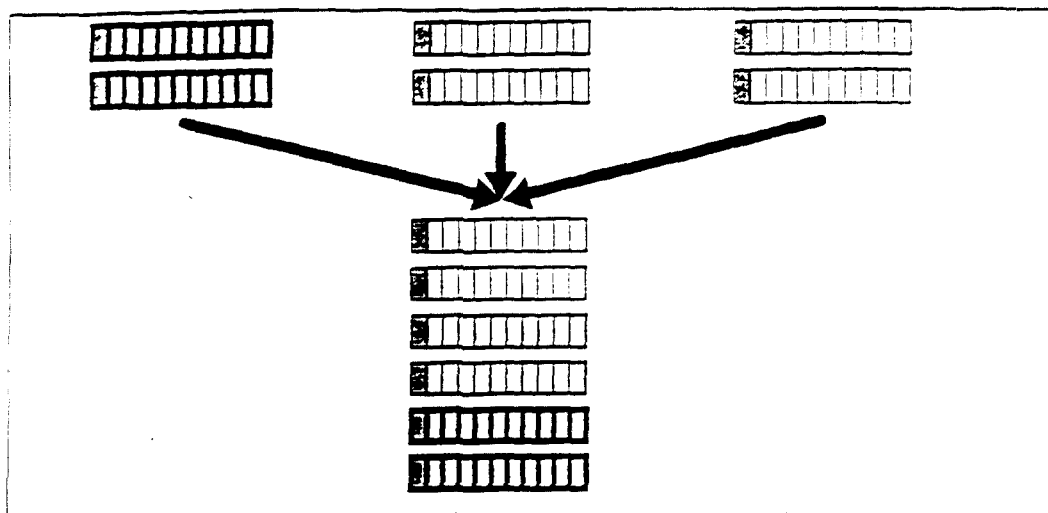


figure 11: Multiplexing

6.2.3.6 Access to LLC

The T-KE shall use the LLC-service assigned in the FlowControl parameter of the T-ASDU. The FlowControl parameter shall be interpreted as defined in 0. For the INITIALISATION.request service the 'DL-UNITDATA.request with response request' service shall be used, for the INITIALISATION.response service the 'DL-UNITDATA.request without response request' service shall be used.

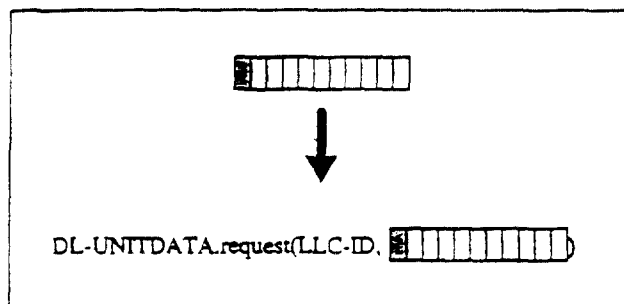


figure 12: Access to LLC

6.2.3.7 Concatenation

Multiple consecutive T-APDU-fragments may be mapped on one LLC-service, if the services and the LID used in the services is the same and the size constraints for the LLC-frames are not violated. The order of the T-APDU-fragments inside the LSDU shall be the one given by the multiplexing procedure.

NOTE: This conditions for concatenation will only allow concatenation in the following two situations:

- 1) One or more T-APDU-fragments, each containing one short, unfragmented T-APDU, are mapped on one LLC-service.
- 2) One or more T-APDU-fragments, each containing one short, unfragmented T-APDU, are mapped on one LLC-service after the last fragment of a series of T-APDU-fragments belonging to one T-APDU

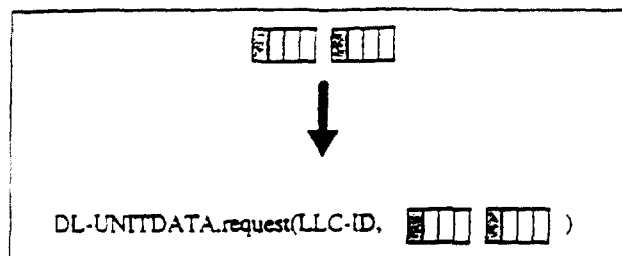


figure 13: Concatenation

6.2.3.8 Demultiplexing

The T-KE shall demultiplex the T-APDU-fragments received in the data fields of the LLC indication primitives according to the PDU number in the fragmentation header. Concatenated fragments shall be demultiplexed according to the PDU number in the first fragmentation header.

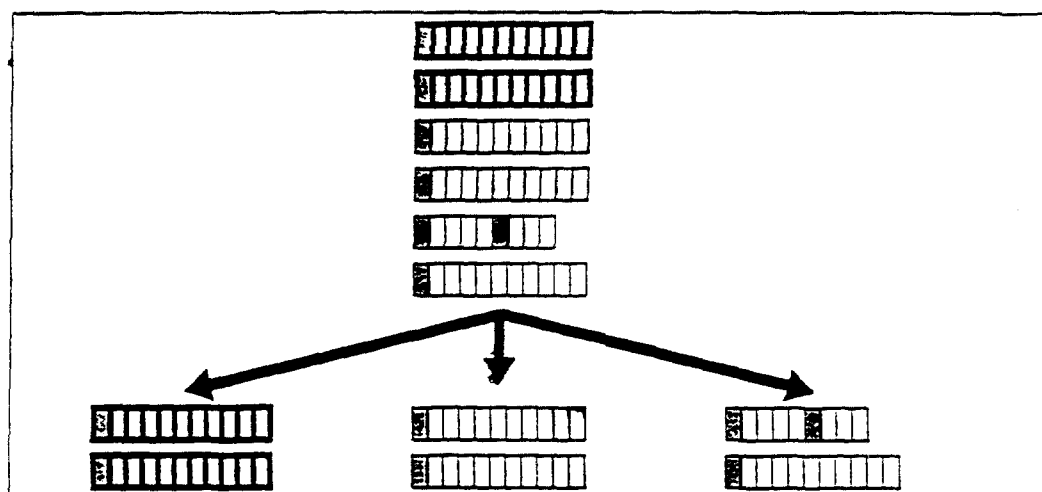


figure 14: Demultiplexing

6.2.3.9 Defragmentation

The T-KE shall defragment the T-APDU-fragments belonging to one T-APDU by removing the fragmentation header and concatenating the T-APDU-fragments according to the fragment number given in the fragmentation header. If the fragmentation header is not valid the fragment shall be discarded. For concatenated fragments the header is not removed and the concatenated T-APDU remains concatenated to the predecessor T-APDU.

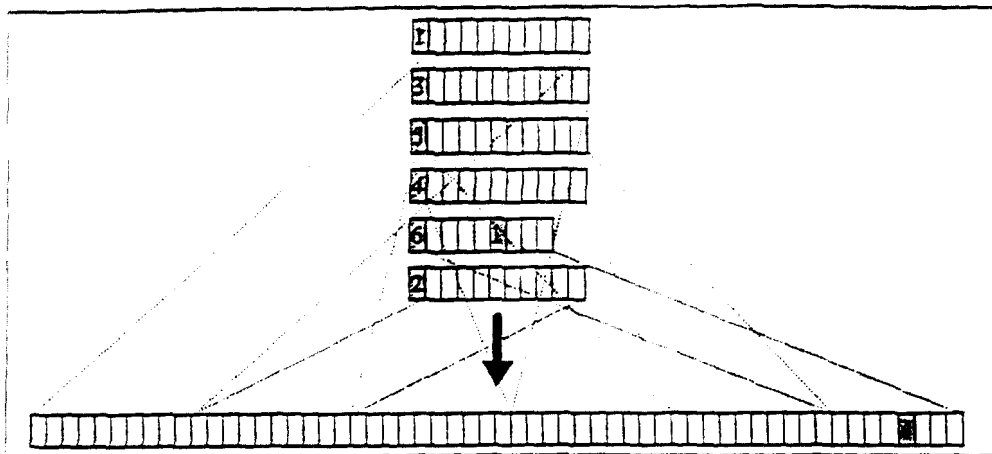


figure 15: Defragmentation

6.2.3.10 Decoding

The T-KE shall decode the defragmented T-APDU according to ASN.1-BASIC-PER, UNALIGNED [ISO 8825-2]. The decodable ASN.1 types are specified in Annex A1. 1 to 7 trailing 'zero' bits shall not result in an error.

If more than 7 trailing bits are received and if not more than 7 bits in sequence have the value zero, these 'zero' bits are removed (as those bits have been inserted to achieve octet alignment). If bit 3 to 1 of the following byte have the value 001₂ this byte shall be removed. The remaining bits shall be decoded as T-APDU according to this clause.

In all other cases the defragmented T-APDU shall be discarded.

If the T-KE is not able to decode the defragmented T-APDU the defragmented T-APDU shall be discarded. The T-KE shall not decode the dummies defined in the container definitions in Annex A.

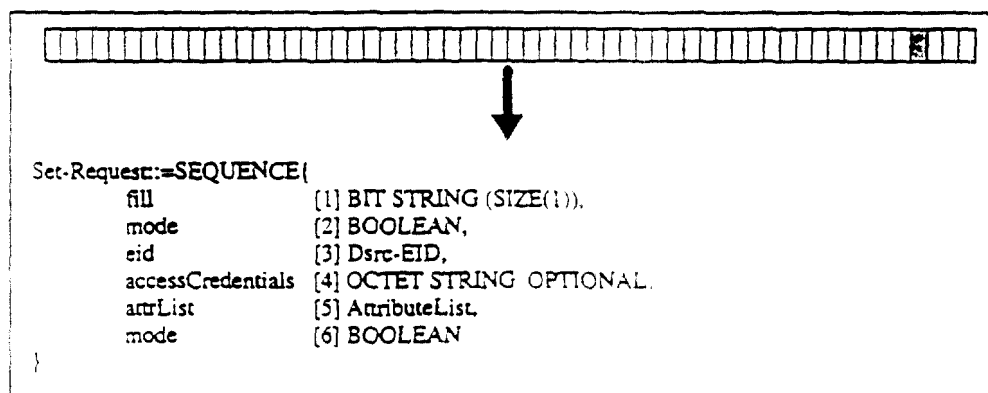


figure 16: Decoding

6.2.3.11 PDU to SDU

The decoded T-APDU shall be used to build the T-ASDU according to the following rules:

A service-request shall be translated into the corresponding service.indication T-ASDU.
A service-response shall be translated into the corresponding service.confirm T-ASDU.

The T-ASDU shall be delivered to the Element addressed in the EID parameter of the T-APDU, but not to the T-KE itself. The INITIALISATION.indication-SDU shall be delivered to the I-KE.

If the addressed Element is not present the T-ASDU shall be discarded.

The T-KE shall inform the Management about the LID of this SDU.

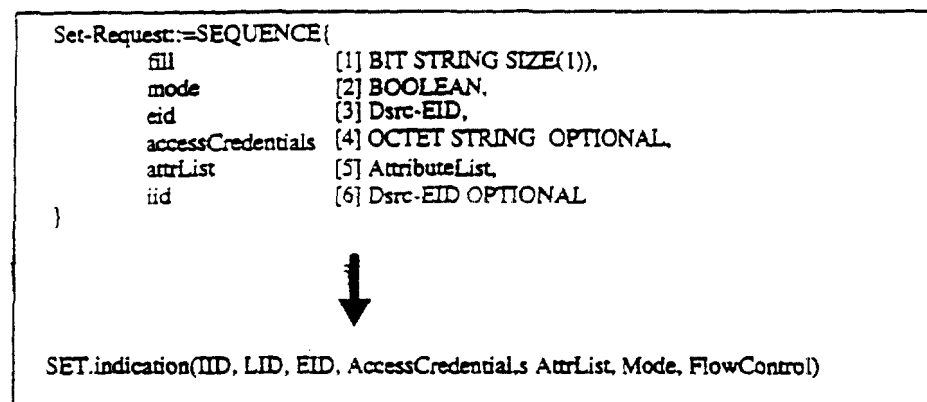


figure 17: PDU to SDU

6.3 INITIALISATION-KE (INITIALISATION-SERVICE-PROVIDER)

6.3.1 SCOPE

The I-KE shall realize the initialisation of the communication between OBE and beacon by exchanging information concerning profiles and applications with its peer entity. It shall inform the applications inside the OBE about the presence of a peer application inside the beacon. It shall handle the LID of the vehicle.

The I-KE shall offer its services by means of service primitives defined in 6.3.2.

The I-KE shall initialise the communication by means of a BST as defined in Annex A.1. The size of one BST shall enable the transfer of the BST in one LLC-Service-Primitive.

The I-KE may initialise the communication by means of a VST as defined in Annex A.1.

The I-KE shall realize the initialisation by a protocol with the behaviour defined in 6.3.3.

The I-KE shall realize associations by means of the LID, the BST and a release mechanism.

6.3.2 SERVICES

6.3.2.1 Scope

The I-KE shall provide the following services to other elements (Initialisation-service-users):

- **RegisterApplicationBeacon:** The invocation of the RegisterApplicationBeacon-service by a service-user (i.e. application on the beacon side) shall result in the notification of potential communication partners about the presence of the service-user (i.e. application on the beacon side).
- **RegisterApplicationVehicle:** The invocation of the RegisterApplicationVehicle-service by a service-user (i.e. application) shall result in the notification of the service-user about the presence of a potential communication partner (i.e. application).
- **DeregisterApplication:** The invocation of the DeregisterApplication-service by a service-user (i.e. application) shall result in the fact that potential communication partners are no longer notified about the presence of the service-user (i.e. application) resp. that the service-user is no longer informed about the presence of a potential communication partner (i.e. application).
- **NotifyApplicationBeacon:** The invocation of the NotifyApplication-service by the service-provider (i.e. beacon's I-KE) shall result in the notification of the service-user about the presence of a potential communication partner (i.e. application) and the LID of the associated OBE.
- **NotifyApplicationVehicle:** The invocation of the NotifyApplication-service by the service-provider (i.e. OBE's I-KE) shall result in the notification of the service-user about the presence of a potential communication partner (i.e. application on the beacon side) and the LID generated by the OBE.
- **ReadyApplication:** The invocation of the ReadyApplication-service by the service-user shall result in the notification of the initialisation-service-provider that the LID is not longer needed for this application. If all applications which are notified invoke the ReadyApplication-service the association between vehicle and beacon will be released by means of invocation of EVENT-REPORT(release) request service primitive.

6.3.2.2 Format of service primitives

The I-ASDU for the service primitives shall have the following format:

RegisterApplicationBeacon:

parameter name	ASN.1 type	optional or default
AID	DSRCApplicationEntityID	
Mandatory	BOOLEAN	
Priority	INTEGER	
EID	Dsrc-EID	optional
Profiles	SEQUENCE OF Profile	optional
Parameter	Container	optional

RegisterApplicationVehicle:

parameter name	ASN.1 type	optional or default
AID	DSRCApplicationEntityID	
Priority	INTEGER	
EID	Dsrc-EID	
Profiles	SEQUENCE OF Profile	optional
Parameter	Container	optional

DeregisterApplication:

parameter name	ASN.1 type
AID	DSRCApplicationEntityID

NotifyApplicationBeacon:

parameter name	ASN.1 type	condition
Priority	INTEGER	
EID	Dsrc-EID	if present for AID in VST
LID	BIT STRING	
Parameter	Container	optional
obeConfiguration	ObeConfiguration	

NotifyApplicationVehicle:

parameter name	ASN.1 type	condition
Beacon	BeaconID	
Priority	INTEGER	
EID	Dsrc-EID	if present for AID in BST
LID	BIT STRING	
Parameter	Container	optional

ReadyApplication:

parameter name	ASN.1 type	condition
EID	Dsrc-EID	
LID	BIT STRING	

6.3.2.3 Parameters

The parameters shall be interpreted as follows:

- AID shall be the DSRCAApplicationEntityID (see Annex A.1) of the Initialisation-service-user (application).
- Mandatory shall be the BOOLEAN status of the Initialisation-service-user (i.e. application). It shall be true if the Initialisation-service-user is a mandatory application and false if the Initialisation-service-user is a non-mandatory application.
- Priority shall be the priority of the Initialisation-service-user in relation to the other Initialisation-service-users. A small INTEGER shall represent a high priority, a high INTEGER a low priority. The Initialisation-service-provider may take this parameter into account when deciding on the priority of the Initialisation-service-user (i.e. application).
- EID may be the Dsrc-EID (see Annex A.1) of the Initialisation-service-user for the RegisterApplicationBeacon-service. If the Initialisation-service-user is not the default Initialisation-service-user (defined by the authority defining the performing Transfer-service-user) this ID shall be set.
- EID shall be the Dsrc-EID of the Initialisation-service-user for the RegisterApplicationVehicle-service.
- EID shall be the Dsrc-EID of the peer Initialisation-service-user for the NotifyApplication-service if present for the Initialisation-service-provider.
- Profiles may be a SEQUENCE OF Profile (see Annex A.1) related to the application. The I-KE may use this profiles to build up the BST.

- Parameter may be additional information related to the initialisation with the Initialisation-service-user (i.e. application). If present it shall be the additional information for the NotifyApplication service of I-KE.
- Beacon may be the BeaconID of the beacon which offers the service.
- obeConfiguration shall be an ObeConfiguration (see Annex A.1) describing the configuration and the status of the OBE related to the LID given in the NotifyApplicationBeacon. This configuration shall be interpreted by the Initialisation Service User. The value shall be given in the VST and shall be set by the management.

6.3.2.4 Location of services

On beacon side the I-KE shall provide the RegisterApplicationBeacon, DeregisterApplication, NotifyApplicationBeacon and ReadyApplication services. On vehicle side the I-KE shall provide the RegisterApplicationVehicle, DeregisterApplication, NotifyApplicationVehicle and ReadyApplication services.

6.3.3 PROTOCOL

6.3.3.1 Beacon: Periodical transmission of BST

Reason:

Time period since last transmission has been expired.

Behaviour:

On beacon side the I-KE shall transmit the BST defined in Annex A.1. It shall use the INITIALISATION.request-service of the Transfer-service-provider (T-KE) with the following parameter settings:

- Initialisation Parameter = BST

The determination of the length of the period between two transmissions is outside the scope of this Standard.

6.3.3.2 OBE: Reception of BST

Reason:

OBE I-KE receives a BST by means of a INITIALISATION.indication of the Transfer-service-provider with the following parameter settings:

- Initialisation Parameter = BST

If at least one of the following conditions is fulfilled the OBE I-KE shall show the following behaviour:

- the BeaconID differs from the last received BeaconID
- the time between the last received BST and the current received BST exceeds 255 seconds

NOTE: This timer shall enable an OBE to decide whether it has lost contact to a beacon or passes the same beacon a second time without passing another beacon in between.

The time may be measured by the OBE or may be calculated by the time given in the BST.

Behaviour:

The I-KE shall inform the Management about the profile received in the BST. This profile shall be the default profile for the communication.

The I-KE shall choose a random LID according to CEN TC278/N474.

The I-KE shall compare the DSRCApplicationEntityIDs given in the ApplicationLists inside the BST with the registered DSRCApplicationEntityIDs. For all registered DSRCApplicationEntityIDs which are inside the BST it shall:

- add the DSRCApplicationEntityIDs and if present in the related RegisterApplicationVehicle the eid and the parameters into the ApplicationList of the VST
- notify the registered element by means of a NotifyApplicationVehicle with the following parameters :
 - Beacon = BeaconID given in the BST,
 - Priority = position in the mand ApplicationList of the BST for all DSRCApplicationEntityIDs inside the mand ApplicationList or the number of DSRCApplicationEntityIDs in the mand ApplicationList plus the registered priority for all DSRCApplicationEntityIDs inside the nonmand ApplicationList,
 - Link Identifier = LID selected by the I-KE,
 - Parameter received in the BST if present, else empty.

The I-KE shall inform the T-KE via the Management about the priorities of the notified applications.

The I-KE may select a profile out of the profileList of the BST which is supported in the OBE (i.e. which is known by the I-KE) and shall set the profile element of the VST to this profile. If it chooses no profile it shall set the profile of the VST to the profile given in the BST.

The VST is of ASN.1 type VST as defined in Annex A.1.

The I-KE shall transmit the VST. It shall use the INITIALISATION.response-service of the Transfer-service-provider (T-KE) with the following parameter settings:

- Link Identifier = LID chosen according to CEN TC278/N474
- Initialisation Parameter = VST

The I-KE shall store the BeaconID beacon and the time.

The I-KE shall store the ApplicationList of the VST together with the LID as long as the LID is valid.

6.3.3.3 Beacon: Answer to VST

Reason:

Beacon I-KE receives a VST by means of a INITIALISATION.confirm of the Transfer-service-provider with the parameters:

- Link Identifier = LID
- Initialisation Parameter = VST

Behaviour:

For all DSRCAApplicationEntityIDs inside the VST the I-KE shall notify the registered element by means of a NotifyApplicationBeacon with the following parameters:

- Priority = position in the mand ApplicationList of the BST for all DSRCAApplicationEntityIDs inside the mand ApplicationList or the number of DSRCAApplicationEntityIDs in the mand ApplicationList plus the position in the nonmand ApplicationList for all DSRCAApplicationEntityIDs inside the nonmand ApplicationList,
- EID = EID received in the VST if present, else empty,
- LID = LID received in the INITIALISATION.confirm,
- Parameter = Parameter received in the VST if present, else empty,
- ObeConfiguration = ObeConfiguration received in the VST,

The I-KE shall inform the Management about the relation between LID and the profile given inside the VST. This profile shall be used for further communication with the OBE with this LID, i.e. outgoing data shall be sent with this profile and incoming data shall be received with this profile.

The I-KE shall store the ApplicationList of the VST together with the LID given in the INITIALISATION.confirm.

6.3.3.4 Beacon: RegisterApplicationBeacon

Reason:

Invocation by an Initialisation-service-user (application on the beacon side).

Behaviour:

On receiving a RegisterApplicationBeacon primitive the beacon-I-KE shall insert the information given in the primitive into the ApplicationList for mandatory or nonmandatory applications respectively. It may use the information given in the parameters 'mandatory' and 'priority'. If a B-KE is present the priority for this Transfer-service-user is fixed by the I-KE. The profiles may be inserted into the profileList of the BST.

6.3.3.5 OBE: RegisterApplicationVehicle

Reason:

Invocation by an Initialisation-service-user (application).

Behaviour:

On receiving a RegisterApplicationVehicle the vehicle-I-KE shall inspect each incoming BST for the application registered in the primitive. After receiving related data it sends a NotifyApplication to the element identified in the RegisterApplicationVehicle primitive together with the priority, EID of the element on beacon side, and the parameters received in the BST as specified in 6.3.3.2.

6.3.3.6 OBE: DeregisterApplication

Reason:

Invocation by an Initialisation-service-user (application).

Behaviour:

On receiving a DeregisterApplication the I-KE shall stop the inspection of the BST according to the application identified in the DeregisterApplication primitive.

6.3.3.7 Beacon: DeregisterApplication

Reason:

Invocation by an Initialisation-service-user (application).

Behaviour:

On receiving a DeregisterApplication the I-KE shall remove the entry in the BST related to the application identified in the DeregisterApplication primitive.

6.3.3.8 Beacon & OBE: ReadyApplication

Reason:

Invocation by an Initialisation-service-user (application).

Behaviour:

On receiving a ReadyApplication the I-KE shall delete the entry from the stored ApplicationList from the VST (see 6.3.3.2 & 6.3.3.3). If the list is empty the I-KE shall transmit a Release to the peer I-KE. It shall use the EVENT-REPORT request-service of the Transfer-service-provider (T-KE) with the following parameter settings:

- Invoker Identifier= (empty)
- Link Identifier= LID
- Element Identifier = EID (of the application)
- EventType = Release (0)
- EventParameter = (empty)
- Mode = FALSE
- FlowControl = 1

6.3.3.9 Beacon & OBE: Reception of a Release

Reason:

I-KE receives a Release by means of a EVENT-REPORT indication of the Transfer-service-provider with the parameters:

- Invoker Identifier= (empty)
- Link Identifier= LID
- EventType = Release (0)
- EventParameter =(empty)
- Mode = FALSE

Behaviour:

The I-KE shall delete the VST related to the LID. The LID shall no longer be valid.

6.4 BROADCAST-KE (BROADCAST-SERVICE-PROVIDER)

6.4.1 SCOPE

The B-KE shall realize the collection, broadcast and distribution of information for different applications in the OBE and beacon by exchanging the Broadcast Pool.

The B-KE shall offer its services by means of service primitives defined in 6.4.2.

The B-KE shall realize the communication by sending a BP defined in Annex A.1.

The B-KE shall realize the broadcast by a protocol with the behaviour defined in 6.4.2.1.

6.4.2 SERVICES

6.4.2.1 Scope

The B-KE shall provide the following services to other elements (Broadcast-service-users):

- BroadcastData: The invocation of the BroadcastData-service by the service-user on beacon side shall result in the broadcast of information to other Broadcast-service-users on OBE side or in the update of this information.
- GetBroadcastData: The invocation of the GetBroadcastData-service by a Broadcast-service-user shall result in the retrieval of the broadcast data.

6.4.2.2 Format of service primitives

The B-ASDU for the service primitives shall have the following format:

BroadcastData:

parameter name	ASN.1 type
File	NamedFile

GetBroadcastData.request:

parameter name	request	confirm	ASN.1 type
Name	mandatory		FileName
EID	mandatory		Dsrc-EID
File		mandatory	NamedFile

6.4.2.3 Parameters

The parameters shall be set and interpreted as follows:

- File shall be the NamedFile which contains the information which shall be broadcast or retrieved, respectively, from the Broadcast Pool.
- Name shall be the FileName of the file which shall be retrieved from the Broadcast Pool.
- EID shall be the Dsrc-EID of the Element (Broadcast-service-user) invoking the GetBroadcastData-service.

6.4.2.4 Location of services

On beacon side the B-KE shall provide the BroadcastData primitive. On vehicle side the B-KE provides the GetBroadcastData.request and GetBroadcastData.confirm primitives.

6.4.3 BEHAVIOUR

6.4.3.1 Beacon: Transmission of BP

Reason:

Time period since last transmission has been expired.

Behaviour:

On beacon side the B-KE shall transmit the BP defined in Annex A. It shall use the SET.request-service of the Transfer-service-provider (T-KE) periodically with the following parameter settings:

- Invoker Identifier = (empty)
- Link Identifier = 11111111₂
- Element Identifier = EID
- Attribute List = (0, BP)
- Mode = FALSE
- FlowControl = 1

The determination of the length of the period between two transmissions is outside the scope of this Standard.

6.4.3.2 OBE: Receiving the BP

Reason:

OBE B-KE receives the BP by means of a SET indication of the Transfer-service-provider with the parameters

- Invoker Identifier= (empty)
- Link Identifier= 11111111₂
- Attribute List=(0, BP)
- Mode = FALSE

Behaviour:

On OBE side the B-KE shall replace the current value of the BP by the new value of the BP.

6.4.3.3 Beacon: BroadcastData

Reason:

Invocation by an Broadcast-service-user (application).

Behaviour:

The B-KE shall insert the file of the NamedFile File into the content of the BroadcastPool and the FileName into the Directory. If there is a File with the same FileName in the BroadcastPool this File is replaced by the new file.

6.4.3.4 OBE: BroadcastData.request

Reason:

Invocation by an Broadcast-service-user (application).

Behaviour:

The B-KE shall retrieve the File with the given FileName and give it to the Broadcast-service-user with Dsrc-EID EID by means of GetBroadcastData.confirm

7 MANAGEMENT

7.1 SCOPE

The Application Layer shall be responsible for the Management of the DSRC Communication System. This Management consists of providing the Application Layer, Data Link Layer, and Physical Layer in both entities with values for the communication parameters and other information necessary for communication and administration of the communication system.

7.2 PROFILES

7.2.1 SCOPE

Inside the Application Layer two types of profiles shall be distinguished:

- DSRC System Profiles: The DSRC System Profile shall represent characteristics of the communication partners and are described in 7.2.2. These profiles shall be handled by the I-KE as described in 6.3 and 0 and are transmitted inside the BST. The ASN.1 type associated with profile is defined in Annex A.1. The definition of the profiles themselves is outside the scope of this Standard.
- Application Layer Profiles: These profiles shall represent characteristics of the Application Layer and are defined in 7.2.3.

7.2.2 DSRC SYSTEM PROFILES

General DSRC System Profiles shall represent characteristics of the communication partners. These characteristics shall be the same or compatible for two communicating partners. Two classes of characteristics shall be distinguished:

- setable characteristics: These characteristics shall be parameters of the system which may be set to a special value in general in each system. They shall be distinguished between characteristics where the wrong setting will lead to communication errors or will disable the communication between the partners and characteristics where the wrong setting will not lead to a communication error or disable the communication between the partners.

NOTE: Timers and Counters which are needed for medium access in the data link layer are examples of setable characteristics with impact on performance in general, but not necessarily on communication errors. Data rates for the Physical Layer is a characteristic which has a direct impact on communication errors.

- abilities: These characteristics shall be abilities of the systems which are present or not. If one communication partner uses this ability the other partner shall also have this ability to be able to understand the incoming data.

NOTE: ASEs or functionalities of ASEs are examples of abilities.

7.2.3 APPLICATION LAYER PROFILES

The Application Layer Standard defines in its current form the ability application layer profiles 0, 1, 2, and 3. New application layer profiles may be registered according to the procedures described in Annex A.2.

The application layer profiles shall be indicated by the 'application layer profile' parameter which describes the abilities of the Application Layer. This parameter shall be used to define profiles that support the DSRC system profiles.

Application Layer Profile number	Characteristics	Intention
0	Transfer Kernel Element with one octet fragmentation and one octet defragmentation Initialisation Kernel Element with ASN.1 Module DSRCtransferData-P0	Identification and simple Interaction
1	Transfer Kernel Element Initialisation Kernel Element with ASN.1 Module DSRCtransferData-P0	Interaction
2	Transfer Kernel Element Broadcast Kernel Element with ASN.1 Module DSRCtransferData-P1	Broadcast
3	Transfer Kernel Element Initialisation Kernel Element Broadcast Kernel Element with ASN.1 Module DSRCtransferData-P2	Interaction and Broadcast

NOTE: The compatibility relations between the different profiles is illustrated in figure 18.

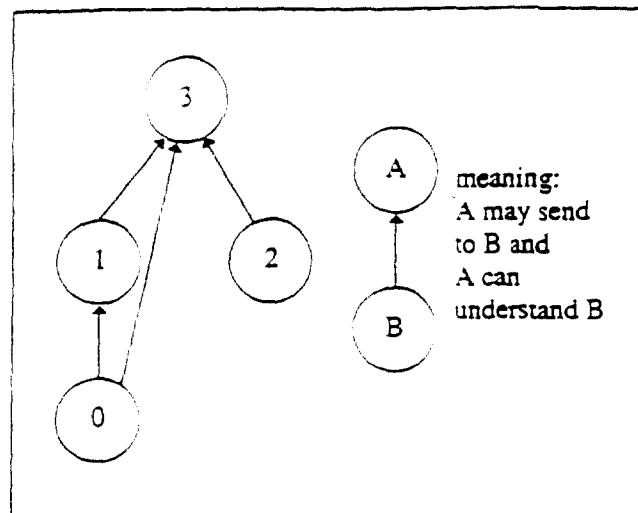


figure 18: Relation between profiles

7.2.4 DSRC SYSTEM PROFILE HANDLING

The DSRC System Profile handling shall consist of the following two phases:

- negotiation phase: The communication partners negotiate a DSRC System Profile for further communication.
- maintenance phase: The roadside communication partner may support different DSRC System Profiles for different mobile communication partners.

7.2.4.1 Negotiation Phase

This phase is described in 6.3. The Management shall store the relation between LID and profile on beacon and on vehicle side. The answer on a BST shall be given with the DSRC System Profile indicated in the BST.

7.2.4.2 Maintenance Phase

After the negotiation phase the negotiated DSRC System Profile shall be used for the transmission to the communication partner.

On vehicle side the communication system shall be able to support the negotiated DSRC System Profile and possible DSRC System Profile for broadcasts in parallel.

On beacon side the communication system shall switch the DSRC System Profile for each incoming message from or outgoing message to a given mobile communication partner. The LLC shall inform the Management about the LID of the next outgoing or incoming data. The Management informs the other layers about the related profile.

7.2.5 INTERLAYER MANAGEMENT AND PRIORITY HANDLING

The I-KE shall inform the Management about the priorities received inside the BST. This information is given to the T-KE for multiplexing purposes.

ANNEX A: DATA STRUCTURES (NORMATIVE)

A 1 USE OF MODULES

The T-KE shall use a DSRCData-Px- and a DSRCtransferData-Px-ASN-1-module where x shall be the number of the used profile.

NOTE: The IMPORT resp. EXPORT mechanism is standardized in [ISO 8824-1].

A 2 ASN.1-MODULES

A.2.1 MODULES FOR PROFILE 0 AND 1

DSRCData-P0 DEFINITIONS ::= BEGIN

IMPORTS

ContainerJ.y FROM ApplicationJ -- this line shall be given for each application
 -- which defines data of type container, J and y
 -- shall be replaced by an unambiguous suffix

Container ::= CHOICE {

integer	[0] INTEGER,
bitstring	[1] BIT STRING,
octetstring	[2] OCTET STRING,
universalString	[3] UniversalString,
beaconId	[4] BeaconID,
t-apdu	[5] T-APDUs,
dsrcApplicationEntityId	[6] DSRCApplicationEntityID,
dsrc-Ase-Id	[7] Dsrc-EID,
attrIdList	[8] AttributeIdList
attrList	[9] AttributeList
dummy	[10..127]NotUsed,

. contI.x [i] ContainerI.x -- this line shall be given for each imported
 -- ContainerI.x, where I.x is replaced by the related
 -- suffix and i is the registered tag starting with 0.
 -- Gaps shall be filled with contI.x [i] BIT STRING

}

-- at this place all ASN.1 type definitions assigned in table 1 shall be inserted. --

END

DSRCtransferData-P0 DEFINITIONS::= BEGIN

IMPORTS T-APDU's FROM DSRCData-P0

Message::= T-APDU's

END

DSRCDData-P1 DEFINITIONS::= BEGIN

ContainerJ.y FROM ApplicationJ

- this line shall be given for each application
- which defines data of type container, J and y
- shall be replaced by an unambiguous suffix

- this line shall be given for each application

- which defines data of type record, J and y
- shall be replaced by an unambiguous suffix

integer

[0] INTEGER.

bitstring

[1] BIT STRING.

octetstring

[2] OCTET STRING,

universalString

[3] UniversalString,

beaconId

[4] BeaconID.

t-apdu

[5] T-APDUs.

dsrcApplicationEntityId

[6] DSRCAApplicationEntityID,

dsrc-Ase-Id

[7] Dsrc-EID.

attrIdList

[8] AttributeIdList

arrList

[9] AttributeList

broadcastPool

[10] BroadcastPool.

directory

[11] Directory,

file

[12] File.

fileType

[13] FileType,

record

[14] Record,

time

[15] Time.

vector

[16] SEQUENCE (0..255) OF
INTEGER(0..127,...),

dummy

[17..127]ReservedForFutureCENUse.